

## CLAIMS

1. A software architecture comprising:  
a manager object capable to  
receive a client request,  
select an adapter object from a plurality of adapter object types  
that is appropriate for handling the client request, and  
delegate the client request to the selected adapter object; and  
a plurality of adapter objects, each adapter object capable to  
receive the client request if the adapter object is appropriate for  
handling the client request,  
select an application component from a plurality of application  
component types that is appropriate for processing the client request, and  
delegate the client request to the selected application  
component.
2. The software architecture of claim 1, wherein the manager object  
is further capable to retrieve data from a manager configuration that is used to  
select the adapter object that is appropriate for handling the client request.
3. The software architecture of claim 1, wherein the adapter object is  
further capable to retrieve data from an adapter configuration that is used to  
select the application component that is appropriate for processing the client  
request.
4. The software architecture of claim 1, wherein the software  
architecture is implemented on an object-oriented programming platform.
5. The software architecture of claim 4, wherein the manager object  
is an instantiation of a manager class and the plurality of adapter objects are  
instantiations of one or more adapter classes.
6. The software architecture of claim 4, wherein the object-oriented  
programming platform is Java or Microsoft .net.

7. The software architecture of claim 1, wherein the manager object comprises a first manager object that handles navigation functionality, a second manager object that handles application functionality, and a third manager object that handles persistence functionality.

8. The software architecture of claim 1, wherein the plurality of adapter objects comprise a first set of adapter objects that handle navigation functionality, a second set of adapter objects that handle application functionality, and a third set of adapter objects that handle persistence functionality.

9. The software architecture of claim 7, wherein the manager object further comprises a fourth manager object that handles logging functionality, a fifth manager object that handles application state management functionality, a sixth manager object that handles data marshalling functionality, and a seventh manager object that handles key management functionality.

10. The software architecture of claim 8, wherein the plurality of adapter objects further comprise a fourth set of adapter objects that handle logging functionality, a fifth set of adapter objects that handle application state management functionality, a sixth set of adapter objects that handle data marshalling functionality, and a seventh set of adapter objects that handle key management functionality.

11. A software architecture comprising:  
a manager class defining methods for  
    receiving the client request,  
    retrieving data from a manager configuration,  
    selecting an adapter class from a plurality of adapter class types  
based on the data from the manager configuration and the client request,  
    instantiating an adapter object from the selected adapter class, and  
    delegating the client request to the selected adapter object; and  
a plurality of adapter classes, each adapter class defining methods for

implementing one or more methods obtained from an adapter interface,

receiving a client request,

retrieving data from an adapter configuration,

selecting an application component class from a plurality of application component class types based on the data from the adapter configuration and the client request,

instantiating an application component from the selected application component class, and

delegating the client request to the selected application component.

12. The software architecture of claim 11, further comprising:  
a manager configuration that includes data specifying which adapter class to select based on the client request; and  
an adapter configuration that includes data specifying which application component class to select based on the client request.

13. The software architecture of claim 11, further comprising:  
an adapter interface defining methods that enable a manager object instantiated from the manager class to exchange data with an adapter object instantiated from one of the plurality of adapter classes; and  
an application component interface defining methods that enable an adapter object instantiated from one of the plurality of adapter classes to exchange data with an application component instantiated from an application component class.

14. The software architecture of claim 11, wherein the software architecture is implemented on an object-oriented programming platform.

15. The software architecture of claim 14, wherein the object-oriented programming platform is Java or Microsoft .net.

16. The software architecture of claim 11, wherein the manager class comprises a first manager class that handles navigation functionality, a second manager class that handles application functionality, and a third manager class that handles persistence functionality.

17. The software architecture of claim 11, wherein the plurality of adapter classes comprise a first set of adapter classes that handle navigation functionality, a second set of adapter classes that handle application functionality, and a third set of adapter classes that handle persistence functionality.

18. The software architecture of claim 16, wherein the manager class further comprises a fourth manager class that handles logging functionality, a fifth manager class that handles application state management functionality, a sixth manager class that handles data marshalling functionality, and a seventh manager class that handles key management functionality.

19. The software architecture of claim 17, wherein the plurality of adapter classes further comprise a fourth set of adapter classes that handle logging functionality, a fifth set of adapter classes that handle application state management functionality, a sixth set of adapter classes that handle data marshalling functionality, and a seventh set of adapter classes that handle key management functionality.

20. The software architecture of claim 12, wherein the manager configuration and the adapter configuration are designed to be modified by a user.

21. The software architecture of claim 11, wherein the plurality of application component class types is provided by a user.

22. The software architecture of claim 11, wherein the manager class defines a functional method relevant to the client request that can be delegated to the selected application component.

23. A method for providing application functionality to a client comprising:
- receiving a client request with a manager object;
  - selecting an adapter object from a plurality of adapter object types that is appropriate for handling the client request;
  - delegating the client request from the manager object to the selected adapter object;
  - selecting an application component from a plurality of application component types that is appropriate for processing the client request;
  - delegating the client request from the adapter object to the selected application component to be processed;
  - receiving application functionality back from the selected application component; and
  - providing the application functionality to the client.
24. The method of claim 23, further comprising retrieving data from a manager configuration that specifies which adapter class to select based on the client request.
25. The method of claim 23, further comprising retrieving data from an adapter configuration that specifies which application component to select based on the client request.
26. The method of claim 23, further comprising:
- providing a manager class that defines methods for receiving a client request, selecting an adapter class, and delegating the client request; and
  - instantiating the manager class to provide the manager object.
27. The method of claim 23, further comprising:
- providing an adapter class that defines methods for receiving a client request, selecting an application component class, and delegating the client request; and
  - instantiating the adapter class to provide the adapter object.

28. The method of claim 23, further comprising:  
implementing one or more methods obtained from an adapter interface in the adapter object, wherein the methods enable the manager object to exchange data with the adapter object.

29. The method of claim 26, wherein providing a manager class comprises providing a first manager class that handles navigation functionality, a second manager class that handles application functionality, and a third manager class that handles persistence functionality.

30. The method of claim 29, wherein providing a manager class further comprises providing a fourth manager class that handles logging functionality, a fifth manager class that handles application state management functionality, a sixth manager class that handles data marshalling functionality, and a seventh manager class that handles key management functionality.

31. The method of claim 27, wherein providing an adapter class comprises providing a first set of adapter classes that handle navigation functionality, a second set of adapter classes that handle application functionality, and a third set of adapter classes that handle persistence functionality.

32. The method of claim 31, wherein providing an adapter class further comprises providing a fourth set of adapter classes that handle logging functionality, a fifth set of adapter classes that handle application state management functionality, a sixth set of adapter classes that handle data marshalling functionality, and a seventh set of adapter classes that handle key management functionality.

33. The method of claim 23, wherein the client request is received from a user.

34. The method of claim 23, wherein the client request is received from a software application.

35. The method of claim 23, wherein the client request is received from another adapter object.

36. The method of claim 23, wherein the client request is received from another application component.

37. The method of claim 26, wherein the manager class defines a functional method relevant to the client request, further comprising:

delegating the functional method from the manager object to the selected adapter object, and

delegating the functional method from the adapter object to the selected application component.

38. A computer program product, physically stored on a machine-readable medium, for providing application functionality to a client, comprising instructions operable to cause a programmable processor to:

receive a client request with a manager object;

select an adapter object from a plurality of adapter object types that is appropriate for handling the client request;

delegate the client request from the manager object to the selected adapter object;

select an application component from a plurality of application component types that is appropriate for processing the client request;

delegate the client request from the adapter object to the selected application component to be processed;

receive application functionality back from the selected application component; and

provide the application functionality to the client.

39. The computer program product of claim 38, further comprising instructions operable to cause a programmable processor to retrieve data from a manager configuration that specifies which adapter object to select based on the client request.

40. The computer program product of claim 38, further comprising instructions operable to cause a programmable processor to retrieve data from an adapter configuration that specifies which application component to select based on the client request.

41. The computer program product of claim 38, further comprising instructions operable to cause a programmable processor to implement methods obtained from an adapter interface in the adapter object, wherein the methods enable the manager object to exchange data with the adapter object.

42. The computer program product of claim 38, further comprising:  
a manager class that defines methods for receiving a client request, selecting an adapter class, and delegating the client request; and  
instructions operable to cause a programmable processor to instance the manager class to provide the manager object.

43. The computer program product of claim 38, further comprising:  
an adapter class that defines methods for receiving a client request, selecting an application component class, and delegating the client request; and  
instructions operable to cause a programmable processor to instance the adapter class to provide the adapter object.

44. The computer program product of claim 42, wherein the manager class defines a functional method relevant to the client request, further comprising instructions operable to cause a programmable processor to:  
delegate the functional method from the manager object to the selected adapter object, and  
delegate the functional method from the adapter object to the selected application component.

45. A data processing system comprising:  
means for receiving a client request with a manager object;



means for selecting an adapter object from a plurality of adapter object types that is appropriate for handling the client request;

means for delegating the client request from the manager object to the selected adapter object;

means for selecting an application component from a plurality of application component types that is appropriate for processing the client request;

means for delegating the client request from the adapter object to the selected application component to be processed;

means for receiving application functionality back from the selected application component; and

means for providing the application functionality to the client.

46. The system of claim 45, further comprising means for retrieving data from a manager configuration that specifies which adapter class to select based on the client request.

47. The system of claim 45, further comprising means for retrieving data from an adapter configuration that specifies which application component to select based on the client request.

48. The system of claim 45, further comprising means for implementing one or more methods obtained from an adapter interface in the adapter object, wherein the methods enable the manager object to exchange data with the adapter object.

49. A software architecture comprising a framework for an application program to interact with a technical infrastructure built on a Java 2 Enterprise Edition platform.

50. The software architecture of claim 49, further comprising structural code that substantially separates software application concerns from technical infrastructure concerns.

51. The software architecture of claim 49, further comprising structural code that substantially separates application components from technical components.